# Curso básico de GAMS
# Verano Cinetífico 2007

Gabriela Garcia Ayala
(mg_garcia@yahoo.com)

## Historia

GAMS (General Algebraic Modeling System) empieza como un proyecto de investigación en el Banco Mundial en 1976. Es pionero en modelación algebraica y se lanza como software comercial en 1987 con
oficinas en Washington, DC y Colonia.

GAMS se utiliza en una gran variedad de áreas como los son: ingeniería agrícola, ingeniería química, econometría, finanzas, economía ambiental, energía, logística, matemáticas, física, comercio internacional, micro/macro economía, investigación de operaciones, etc.

GAMS es un sistema de modelación ampliamente utilizado. De febrero 11 a marzo 12 del año 2007 se bajaron 1963 programas de su página. Esto equivale a aproximadamente a 500 programas por semana.[i]

## Sistemas de Modelación

Existen diferentes sistemas para modelación matemática, sus principales características son:

1) Sistemas de modelación algebraica → modelos de ecuaciones puras
2) Capacidad de manejar índices → optimización a gran escala
3) Diferenciación automática → no hay que ingresar derivadas
4) Interfase automática con los resolvedores

Algunos ejemplos de sistemas de modelación comerciales para programación matemática son: GAMS, AMPL, AIMSS

## Características de un programa GAMS

✓ Un programa de GAMS es un archivo de texto con terminación .gms. Esto quiere decir que se puede escribir en cualquier procesador de texto.

✓ Hay diferentes formas de correr un programa en GAMS, en el laboratorio sólo hay que escribir "/gams ", en una terminal seguido del nombre del archivo con el programa.
(por ejemplo: gams mimodelo.gms)

✓ Después de correr un modelo, GAMS creará un archivo con el mismo nombre del programa, pero con terminación *.lst en el mismo directorio donde se encuentre el programa.
(por ejemplo: al correr mimodelo.gms, se creará mimodelo.lst con los resultados)

✓ Se puede utilizar cualquier tipografía, número de espacios, o de renglones

✓ GAMS no distingue entre letras mayúsculas y minúsculas

✓ Los renglones que principien con un asterisco "*", en la primer columna son considerados como comentario

✓ La única regla en cuanto a orden en GAMS es que no se puede utilizar un argumento no sea declarado con anticipación

✓ El punto y coma ";" indica el final de cualquier operación, ya sea de declaración o asignación

✓Los nombres de las entidades en GAMS deben empezar con una letra y pueden ser de hasta nueve caracteres

✓ Se le puede pedir a GAMS el mandar resultados específicos de interés como puede ser formatos de salida, datos intermedios de en las iteraciones, etc.

✓ Se puede especificar que solver utilizar, e incluso adicionar un archivo con parámetros para el solver como número de iteraciones, o de decimales a tomar en cuenta, datos de las derivadas, entre otros.

✓ En la página de GAMS en *gams.com* , se puede encontrar el manual del usuario, así como diversos tutoriales y una EXTENSA librería de programas para modelos de todas índoles de la optimización.

✓ Dentro de la documentación disponible, se encuentran los manuales de cada solver, y las formas en que se puede interactuar con ellos.

## Estructura de un modelo GAMS

```
$TITLE Test Problem

*Assignment problem for heat exchangers from pp.409-410 in "Optimization
* of Chemical Processes" by Edgar and Himmelblau

SETS
        I       corrientes /A, B, C, D/
        J       intercambiadores /1*4/;

TABLE C(i,j) Costo de Asignarle a la corriente i el intercambiador j
        1       2       3       4
A       94      1       54      68
B       74      10      88      82
C       73      88      8       76
D       11      74      81      21;

VARIABLES  X(I,J)  ,   Z;
BINARY VARIABLES X(i,j);

EQUATIONS
        ASI(J), ASJ(I), OBJ;

OBJ .. Z =e= SUM(   (i,j),    C(i,J)*X(i,j)
ASI(J).. SUM( I, X(i,j) )=e= 1;
ASJ(I).. SUM(J, X(i,j) )=e= 1;

MODEL HEAT /ALL/;

solve HEAT using MIP  minimizing Z;
```

Declaración y asignacion de parámetros

Declaración de Variables

Definición y declaración de ecuaciones

Armar modelo

Llamar a  solver

## Declaraciones en  GAMS

GAMS maneja cuatro tipos de registros que deben ser declarados:

DATOS *set* i rutas disponibles  /1,2,3,4/;
*Parameter* capacidad(i);
*Table*  distancia (i,j) distancia entre ciudades
*Scalar* f costo por retraso /90/;

VARIABLES *variable* x(i,j),y;
*binary variable* y;
*positive variable* producción(j), inventario;

ECUACIONES  *equations*
costo  funcion objetivo

```
                demanda(j)   demanda en de la ciudad j;
MODELOS    model capacidades /all/;
           model modelo1 /costo,funcion1,funcion2,funcion3/;
```

## Asignaciones en GAMS

```
TABLE C(i,j) Costo de Asignarle a la corriente i el intercambiador j
        1       2       3       4
A       94      1       54      68
B       74      10      88      82
C       73      88      8       76
D       11      74      81      21;
Scalar f, g, z;
f=30;
```

## Asignación o definición de funciones

```
OBJ .. Z =e= SUM(   (i,j),    C(i,J)*X(i,j)   );
ASI(J).. SUM( I, X(i,j)) =e= 1;
ASJ(I).. SUM(J, X(i,j)) =e= 1;
```

## Resolver un modelo en GAMS

Una vez que se ha definido un modelo en GAMS, se puede entonces resolver con un llamado a un solver.

Minimizar o maximizar
solve HEAT using MIP minimizing Z;
Nombre con el que se definó en modelo
Tipo de modelo, en este caso es un MILP, al cual GAMS denomina MIP
Una de las variables decalaradas

El solver que utilizará GAMS será en este caso el que tiene prestablecido por default.
Se puede escoger que solver utilizar, por ejemplo con el siguiente comando:

```
OPTION MIP= cplex;
```

Estoy escogiendo a cplex como mi solver. Dicho renglón deberá colocarse antes de invocar el comando ¨solve¨.

## *Tipos de Modelos en GAMS*

MODELOS LINEALES LP
MODELOS NO-LINEALES NLP
MODELOS MIXTOS LINEALES MIP

MODELOS MIXTOS NO-LINEALES MINLP
Modelos mixtos lineales relajados rMIP
Modelos mixtos no-lineales relajados rMINLP
DNLP Modelo no-lineal con discontinuidades en la derivada
MCP Modelo mixto de complementariedad
CNS Sistema no-lineal restringido

## Compilación en GAMS

Antes de correr el modelo GAMS ejecuta una compilación para localizar errores de sintaxis.
De haber un error de compilación, avisa al usuario que hubo un error, e imprime el archivo de listado con el programa marcando con un número clave el error que ocurrió y en donde fue. En el manual de usuario aparece la lista de errores posibles; aunque por lo general son auto-explicativos.
Resultados en GAMS

Una vez que se corre un programa, GAMS crea un archivo con el mismo nombre del programa, pero con la extensión .lst

El archivo *.lst esta compuesto las siguientes partes:

- Echo print
- Mensajes de Error
- Mapas de Referencia
- Listado de Ecuaciones
- Estadísticas del Modelo
- Reporte de Estatus
- Reporte de Solución

El **echo print**, es solo una versión copiada del modelo que estamos resolviendo. Gams numera los renglones, lo cual es útil para cuando indica errores.

Los **mensajes de error**, aparecen al final del programa del echo, una lista con los errores con su código, indicando que error es y en que renglón se encuentra.

Los **mapas de referencia** indican las variables, su nombre, tipo y referencia de donde se utilizaron.

El **listado de ecuaciones** presenta en orden las ecuaciones utilizadas en el modelo con sus variables en especifico. Esta es la mejor herramienta útil cuando no se están obteniendo resultados esperados.

**Las estadísticas** del modelo se refieren a su tamaño, numero de variables, numero de ecuaciones, etc.

Una vez que el modelo haya corriendo satisfactoriamente , se generan dos reportes. El **reporte de estatus** indica bajo que condiciones se detuvo la corrida. Es decir, si encontró solución, si es única, si no es factible, si llega al limite de memoria o de iteraciones, etc. Hay una lista de estatus en el manual, que indican todas las condiciones bajo las cuales se puede terminar una corrida.

El **reporte de solución** es seguramente lo de mas interés para el usuario puesto que es donde se presentan los resultados de la corrida. Se presentan los resultados primero por ecuación, segundo por variable y por último por petición del usuario.

# Apuntes útiles

**Extensiones** (solo para variables)

.l en el estado actual
.up cota superior
.lo cota inferior
.m multiplicadores del simples

**Operador card y ord** (solo para conjuntos)

set I /1*3/;
ord(i) puede ser 1, 2 o 3
card(i) cardinalidad del conjunto i, en este caso 3

**Operador de dólar ($)**. Este es u operador condicional.
Para definir ecuaciones;

distancia(i,j)$(uso(i) ne 1)..

Dentro de ecuaciones;

costo.. Sum((i,j)$(ord(i) ne ord(j)), variable (i,j)=l= 5

**Comando** $include
Este es un comando muy útil puesto que permite “pegar” lo que existe en un archivo dentro del programa de GAMS.
Por ejemplo si tengo
$include ex4.dat
dentro de un programa, entonces correrá con los datos del ejemplo 4. Se pueden correr varios ejemplos con tan solo cambiar este renglón.

LOOPS
Útiles para cuando sé esta escribiendo un algoritmo por ejemplo:

```
Set i iteration counter /1*30/;
Scalar  UB=inf
        LB = -inf
        count;
Loop (i$((UB-LB) ge 0.001)
        count=ord(i);
```

```
        solve nlpmodel using nlp minimizing nlpobj;
        UB $(nlpobj.l le UB) = nlpobj.l;
        Milp_param(i)=nlp_var.l;
        Solve milpmodel using mip minimizing milpobj;
        LB$(milpobj.l ge LB0 = milpobj.l;
        Nlp_param(i+1)= milp_var.l
);
```

**Comando IF**

Sirve para condicionar ecuaciones o asignaciones.

```
LOOP(i,
    solve nlpmodel using nlp minimizing nlpobj;
    if((nlpobj.l ge milpobj.l),
        solve milpmodel using mip minimizing milpobj);
);
```

# Últimos comentarios

✓Se puede bajar la última versión de GAMS de la página de Internet.

http://download.gams-software.com/

Hay una licencia limitada de libre uso. Los limites para la licencia libre según la pagina de Gams:

1. **Model limits:**
   - Number of constraints and variables: 300
   - Number of nonzero elements: 2000 (of which 1000 nonlinear)
   - Number of discrete variables: 50 (including semi continuous, semi integer and member of SOS-Sets)
2. **Global solver limits:**
   - Number of constraints and variables: 10

✓Los solver que están habilitados en Yalma son:

```
Type        Default     Current
LP          CPLEX
MIP         CPLEX
RMIP        CPLEX
NLP         CONOPT
MCP         MILES
MPEC        NLPEC
DREAM       *
CNS         CONOPT
```

| | | |
|---|---|---|
| DNLP | CONOPT | |
| RMINLP | CONOPT | |
| MINLP | DICOPT | DICOPT |
| QCP | CONOPT | |
| MIQCP | CPLEX | |
| RMIQCP | CONOPT | |

✓Gams cuenta con una dirección de ayuda en donde se pueden mandar dudas.

support@gams.com

Contestan muy rápido. Por favor para un mejor servicio, hacer preguntas especificas!

✓El manual de usuario de GAMS se encuentra tanto en la página de GAMS como en yalma en la siguiente dirección:

http://yalma.fime.uanl.mx/gams-manual/

---

[i] Annual Review Meeting, Center for Advanced Process Decision Making (CAPD) March 2007, CMU Pittsburgh, PA. Jan-H. Jagla, Lutz Westermann; **GAMS: Productivity - Performance - Reliability**.